Introduction

Power BI

Angelo Mammana

Tesina Sistemi informativi e Web Semantico

Overview Power BI

Power BI is a business intelligence platform that allows users to intuitively analyse, visualise and share data. With Power BI, you can import data from various sources, create interactive reports and customised dashboards to gain insight into business performance. Power BI also offers collaboration and sharing tools, allowing users to work together to make data-driven decisions.

Functionality

This system offers a wide range of features that enable users to explore and analyse data effectively. Some of the key features of Power BI include:

- Power Query: allows data to be imported from different sources, transformed and prepared for analysis.
- Power Pivot: allows you to model data and create relationships between tables to generate complex analyses.
- Power View: offers a wide range of interactive data visualisations, including graphs, maps and dynamic tables.
- Power BI Desktop: this is a desktop application that enables the creation and design of reports and dashboards.
- Power BI Service: this is a cloud-based platform for sharing, collaborating and publishing reports.
- Power BI Mobile: allows reports to be accessed and viewed on mobile devices, enabling on-the-go analysis.

Interactive dashboards

Using the Desktop version of the software, it is possible to create dashboards that allow users to explore data intuitively. The process of creating a dashboard starts with importing data from different sources (excel, csv, MySQL..). Power Query allows the data to be transformed, cleaned and prepared for analysis. Then, using Power BI Desktop, interactive data visualisations such as graphs, maps, tables and filters can be created. Users can organise these visualisations in a dashboard, which provides a visual overview of the data.

Power Query

Power Query is a powerful feature that allows you to import, transform and combine data from different sources. You can perform data cleaning and formatting operations, such as removing duplicate rows, filling in missing values or splitting columns. You can also merge several tables or data sources to create a unified view of the data. This functionality offers an easy-to-use interface, allowing users to shape the data according to their needs, without having to write code.

Additional functionalities

Data visualisations are one of the strengths of Power BI. In addition to traditional charts such as bars, towers and lines, Power BI offers a wide range of advanced visualisations, including maps, scatter plots and bubble charts. These visualisations allow users to present data in a way that makes it easier to understand the information. In addition, it allows customisation of visualisations.

Sharing reports is essential to enable users to visualise and interact with data. Power BI offers tools for sharing reports and dashboards. Through the Power BI Service, reports can be shared with colleagues or customers, giving them access to real-time data. However, data security is crucial. Power BI offers advanced options to ensure data security, allowing you to control access to reports and define user permissions.

Automation features are provided to simplify the report generation process and improve efficiency. It is possible to automate the updating of report data by scheduling automatic updates at specific intervals. In addition, Power BI can be integrated with other business applications, such as Microsoft Excel, SharePoint and Teams, to provide a unified view of the data. For example, Power BI reports or dashboards can be embedded in SharePoint pages or Teams tabs, facilitating access to data.

Analysis of sales

Power BI - Basic part

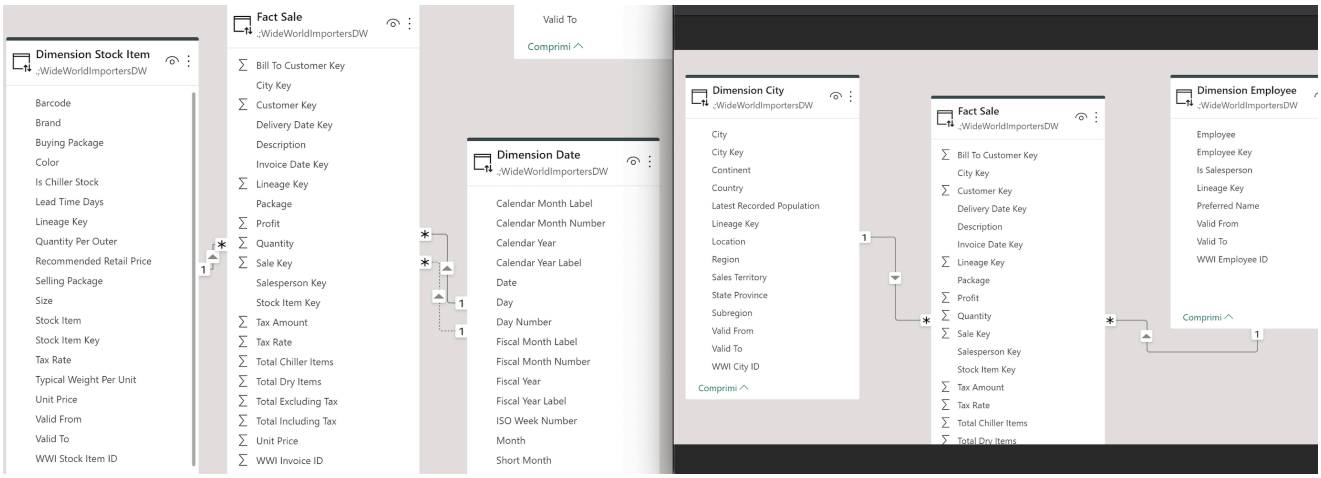
Introduction to dataset

The dataset provided for this analysis was retrieved from the SQL Server sample datasets. Using MySQL, we imported the data from these data sources into our Power BI working environment. This has provided us with a comprehensive set of realistic data to explore and analyze sales, dates, stock items, cities, and employees of a company.

The dataset consists of 5 interconnected tables that allow us to gain a complete understanding of the business operations. The main table, referred to as the "fact table," contains detailed information about sales and related attributes such as customer, delivery date, product, profit, and quantity sold. We will use this table as a starting point for our analyses. Additionally, we have the following tables:

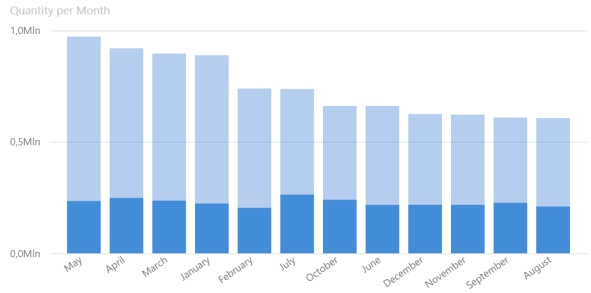
- The "dimension date" table provides details about the timing of sales.
- The "dimension stock item" table contains information about the available stock items.
- The "dimension city" table offers detailed information about the cities involved in the sales.
- The "dimension employee" table contains details about the employees

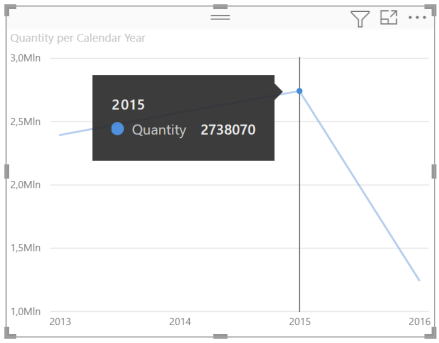
By leveraging these interconnected tables in Power BI, we will be able to gain valuable insights into the sales, analyse the timing of transactions, explore the product inventory, understand the geographical aspects of the business, and assess the role of employees in the company's performance.

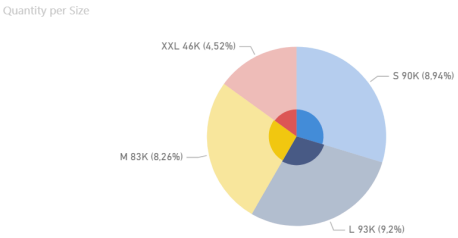


Analysis

The first part of the analysis was very informative, during which it was possible to take the first steps to understand the functionalities of the BI system. Three different visualizations were created to analyze the quantities produced by the company:





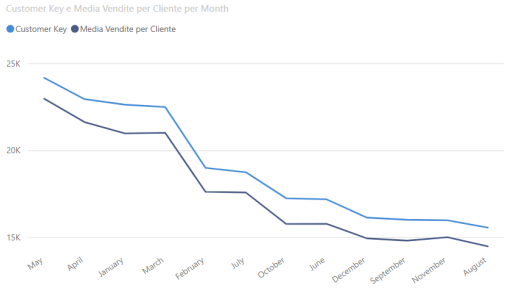


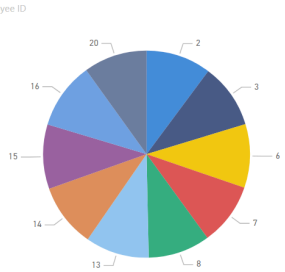
these graphs make us realize that the highest productions occurred in the year 2015, and with the help of the pie and bar charts, we become aware that July was the month with the highest production of size L shirts.

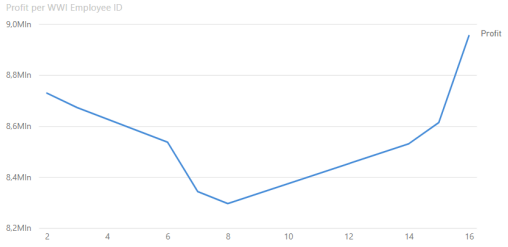
A crucial tool utilized in our work is the DAX language: Data Analysis Expressions. DAX is a formula language used for creating expressions and calculations in our report's data models. Through this language, we have crafted new measures to advance our analyses, notably including the measure "Media vendite per cliente":

```
Media vendite per cliente =
DIVIDE(
SUM('Fact Sale'[Profit]),
DISTINCTCOUNT('Fact Sale'[Customer Key])
)
```

As illustrated in the graph, May stands out as the month with the highest sales.



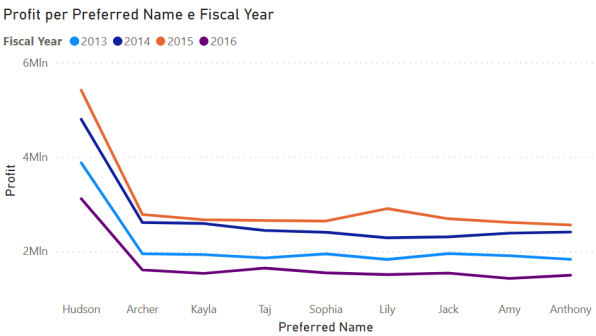




After developing a chart displaying the profits of sellers based on months, we calculated the average difference in profits between the current year and the previous one. Finally, we represented all the data in a matrix.

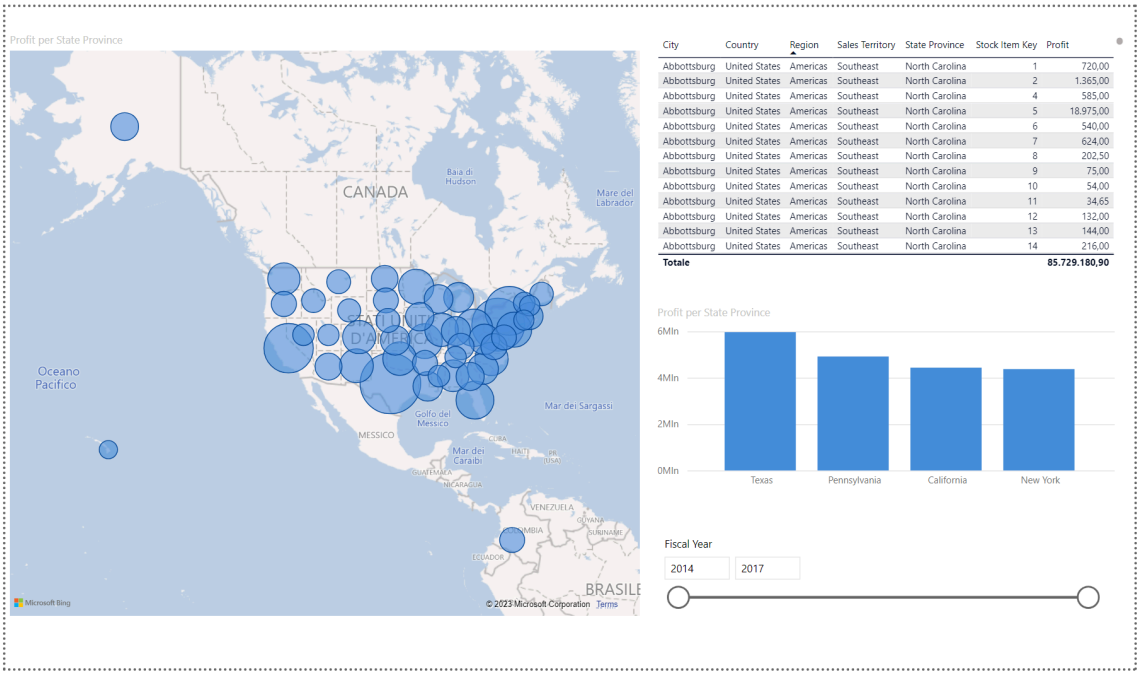
```
Differenza di Profitto tra Anni Fiscali =
VAR CurrentYearProfit = SUM('Fact Sale'[Profit])
VAR PreviousYearProfit = CALCULATE(SUM('Fact Sale'[Profit]),
                    FILTER(ALL('Dimension Date'), 'Dimension Date'[Fiscal Year] =
MAX('Dimension Date'[Fiscal Year]) - 1))
RETURN
CurrentYearProfit - PreviousYearProfit
```

It is evident that Hudson is the seller with the highest profit compared to his competitors. Additionally, we observed a significant profit loss in 2016, unlike the period from 2013 to 2015, where profits were consistently increasing.



Preferred Name	2013	2014	2015	2016	Totale
Amy	1.905.376,10	479.077,75	228.093,80	-1.188.821,25	**5.713.556,35**
Anthony	1.828.987,80	579.008,35	149.870,70	-1.064.785,30	**5.730.065,50**
Archer	1.947.758,50	661.670,10	170.398,35	-1.174.423,55	**6.162.590,50**
Hudson	3.875.311,80	921.142,20	616.808,15	-2.299.005,25	**11.786.022,70**
Jack	1.951.185,55	355.188,15	383.597,15	-1.151.393,90	**5.796.136,20**
Kayla	1.929.768,20	659.561,15	79.347,80	-1.137.102,75	**6.050.671,95**
Lily	1.827.803,45	457.636,50	618.822,10	-1.398.101,80	**5.619.403,65**
Sophia	1.943.747,05	459.087,35	238.229,85	-1.098.932,05	**5.888.713,65**
Taj	1.859.642,80	580.802,20	211.851,50	-1.009.595,20	**5.942.789,10**
Totale	**19.069.581,25**	**5.153.173,75**	**2.697.019,40**	**-11.522.161,05**	**58.689.949,60**

Another aspect we evaluated was to determine in which regions the highest sales were recorded. To provide a more illustrative visualization, I used the mapping tool in the BI system, which clearly depicts the distribution of profits for each state. The resulting data shows that the four highest profits from 2014 to 2017 came from Texas, followed by Pennsylvania, California, and New York.



As a final analysis, we decided to explore the frequent itemsets contained within this dataset. To conduct this market basket analysis of products purchased together, we need to create a new table containing relevant information, such as the sale number and the products purchased in each sale. We will achieve this using the following DAX function:

summarize(Transazioni = SUMMARIZE('Fact Sale', 'Fact Sale'[sale key], 'Dimension Stock Item'[stock item key], 'Fact Sale'[Description]))

This function allows us to create the "Transazioni" table. Subsequently, we will utilize Power BI's Python integration to examine the presence of frequent itemsets. In the code, we import the pandas, combinations, and counter libraries and use them as follows:

Editor di script Python

```
13
14
15 combinations_count = Counter()
16
17 #costruisco le combinazioni di prodotti
18 grouped_data = data.groupby('sale_key')['stock Item key'].apply(list)
19 for products_list in grouped_data:
20     #genera le combinazioni di prodotti per ciascun gruppo
21     product_combinations = combinations(products_list, 2)
22     combinations_count.update(product_combinations)
23
24 print("Prodotti acquistati spesso insieme:")
25 for combination, count in combinations_count.most_common():
26     product1, product2 = combination
27     print(f"{product1} e {product2}: {count} volte")
28
```

Pagina 1 | Pagina 2 | Pagina 3 | Pagina 4 | +

Product1_Product2
(168, 179)
(168, 180)
(168, 185)
(168, 191)
(168, 194)
(168, 199)
(168, 203)
(168, 220)
(172, 168)
(172, 179)
(172, 180)
(172, 185)
(172, 191)
(172, 194)
(172, 199)
(172, 203)
(172, 220)
(179, 180)
(179, 191)
(179, 194)
(179, 220)
(180, 191)
(180, 194)
(180, 220)
(185, 179)
(185, 180)
(185, 191)
(185, 194)

Following the execution of the code, it provided us with the following results, demonstrating that, for instance, item 168 is frequently purchased together with items 179 and 180.

Analysis of sales

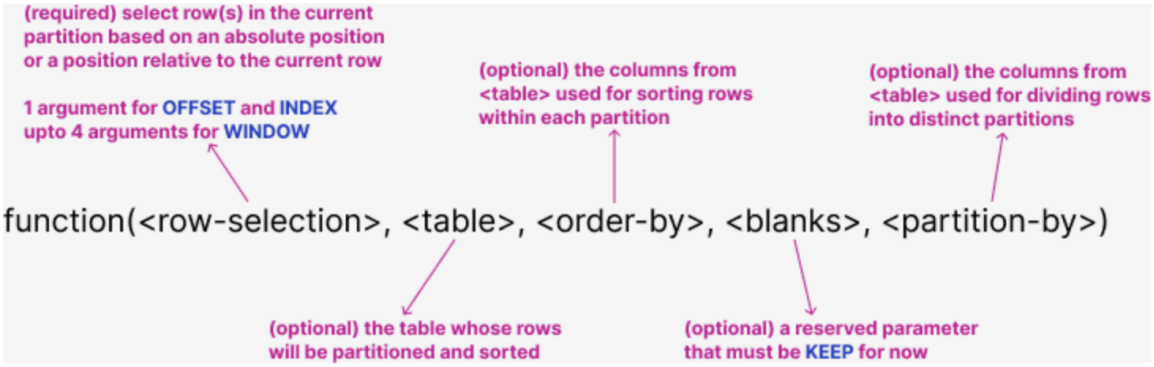
Power BI - Window Function, Partition, OFFSET

Overview Window Function

The Windows function is one of the new DAX functions that were released in 2022. Using WINDOW, a slice of results is retrieved using either absolute or relative positioning. The WINDOW function can be defined using a combination of functions such as FILTER, ALL, and VALUES based on the requirement.

The Window function reads all of the data from the table provided by the <table> parameter and divides the records into distinct divisions using partition-by-columns' unique values. Using the order-by-columns and sorting instructions, arrange the rows within each partition. Returns all the rows between a lower bound and an upper bound, depending on the current partition and the current row.

Instead of being pushed to the data source, DAX functions like Window's are executed within the DAX engine. These DAX functions have shown significantly improved performance compared to existing DAX expressions, particularly when sorting non-continuous columns is necessary.



Analysis of the Average Total Profit over a Period of 3 Months

The first experiment concerns the use of window functions to conduct a temporal analysis within specific time intervals. The calculation uses a WF to examine the time trend of profits during 3-month intervals. Specifically, it calculates the average of profits based on data over the past 3 months. This makes it possible to identify how profits have developed over time during this reference period.

Formula applied:

```
Avg Total profit 3M = AVERAGEX( WINDOW(-2,REL, 0,REL, SUMMARIZE(ALLSELECTED('Dimension Date'),
'Dimension Date'[Fiscal Year],'Dimension Date'[Fiscal Month Number]),ORDERBY('Dimension Date'
[Fiscal Year], ASC, 'Dimension Date'[Fiscal Month Number], ASC)), [Total profit])
```

WINDOW(-2, REL, 0, REL, ...): This part defines a moving window that starts two rows before the current relative context and extends to the current relative context,0, in the date context. This moving window interval represents 3 months.

SUMMARIZE: to create a temporary table that includes the 'Fiscal Year' and 'Fiscal Month Number' columns from the 'Dimension Date' table. The ALLSELECTED function returns all rows selected in the current context.

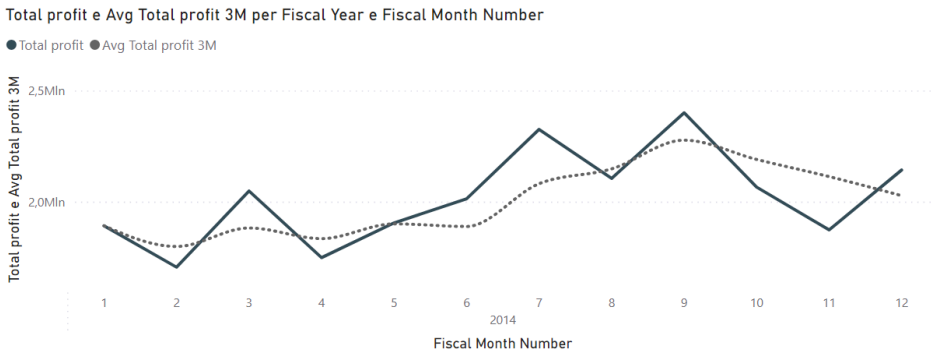
AVERAGEX: to calculate the average 'Total profit'. This means that you are averaging profit only for sales that fall within the 3-month interval defined by the moving window.

Results obtained:

Fiscal Year	Fiscal Month Number	Total profit	Avg Total profit 3M
2013	9	2.119.111,10	2.144.248,13
2013	10	1.758.930,70	1.989.505,00
2013	11	1.856.730,90	1.911.590,90
2013	12	1.880.013,50	1.831.891,70
2014	1	1.891.506,05	1.876.083,48
2014	2	1.705.477,90	1.825.665,82
2014	3	2.048.065,55	1.881.683,17
2014	4	1.748.311,25	1.833.951,57
2014	5	1.904.433,05	1.900.269,95
2014	6	2.013.126,10	1.888.623,47
2014	7	2.325.461,60	2.081.006,92
2014	8	2.104.809,15	2.147.798,95
Totale		**85.609.724,00**	**2.088.042,05**

I dati partono dal Marzo del 2013, per cui il rispettivo AVG Total profit rispetto i 3 mesi di Marzo 2013 rimane il total profit di Marzo.

Dal grafico osservato salta subito all'occhio di come la media calcolata sia superiore a partire dalla seconda metà del 2014 e in particolar modo nel mese di settembre raggiunge il picco medio del profitto



Cumulative calculation by T-shirt color category

A subsequent analysis focuses on a cumulative calculation based on product categories. Since the database lacks defined criteria to distinguish one product from another, I opted to use colors for reference.

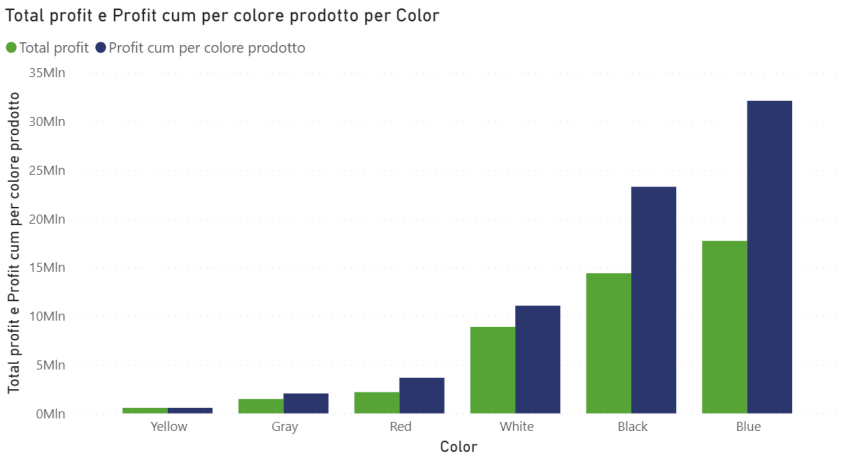
Formula applied:

```
Profit cum per colore prodotto =
CALCULATE([Total profit]
    ,WINDOW(-1, REL, 0, REL,
        SUMMARIZE(
            ALLSELECTED('Dimension Stock Item'),
            'Dimension Stock Item'[Color]
        ),
        ORDERBY(
            [Total profit],ASC

        )
    ))
```

Results obtained:

Color	Total profit	Profit cum per colore prodotto
Yellow	569.302,50	569.302,50
Gray	1.475.748,75	2.045.051,25
Red	2.179.947,50	3.655.696,25
White	8.882.848,50	11.062.796,00
Black	14.390.527,00	23.273.375,50
Blue	17.713.073,00	32.103.600,00
Totale	**45.211.447,25**	**45.211.447,25**



Based on the observations made, the higher profits associated with blue and black T-shirts clearly emerge, in sharp contrast to the alternative colors (such as yellow, gray, red and white) which, taken together, cannot even match the level of profits generated by either option.

Highest Sales Month with Partition by Year

In this other case, the goal of this function is to analyze the highest total sales for each fiscal year by partitioning the data by fiscal year. This is useful to identify the years in which sales were the best performing.

To calculate total sales, first calculate the 'Total Sales' measure using the following formula:
Total Sales = SUMX('Fact Sale', 'Fact Sale'[Unit Price] * 'Fact Sale'[Quantity])

Next, I build the more complex function. Which gives me a detailed overview of sales performance over time.

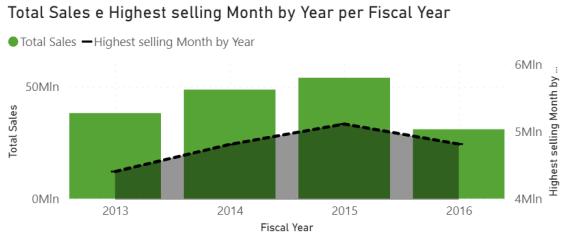
Formula applied:

```
Highest selling Month by Year =
MAXX(
    WINDOW(1, ABS, -1, ABS,
        SUMMARIZE(
            ALLSELECTED('Dimension Date'),
            'Dimension Date'[Fiscal Year], 'Dimension Date'[Fiscal Month Number]
        ),
        , ,PARTITIONBY('Dimension Date'[Fiscal Year])

    ), [Total Sales]
)
```

Results obtained:

Fiscal Year	Fiscal Month Number	Total Sales	Highest selling Month by Year
		232.583,95	232.583,95
2014	1	3.788.913,15	4.807.740,50
2015	1	4.290.230,60	5.111.993,90
2016	1	4.016.720,95	4.813.313,60
2014	2	3.468.811,50	4.807.740,50
2015	2	4.190.135,85	5.111.993,90
2016	2	4.387.639,60	4.813.313,60
2013	3	3.644.848,35	4.404.250,45
2014	3	4.071.710,85	4.807.740,50
2015	3	4.451.461,80	5.111.993,90
2016	3	4.649.480,05	4.813.313,60
2013	4	2.776.929,70	4.404.250,45
2014	4	3.519.594,60	4.807.740,50
2015	4	4.191.734,95	5.111.993,90
2016	4	3.746.196,30	4.813.313,60
2013	5	3.995.924,30	4.404.250,45
2014	5	3.864.591,40	4.807.740,50
2015	5	4.530.291,95	5.111.993,90
2016	5	4.693.816,90	4.813.313,60
Totale		**172.261.341,20**	**5.111.993,90**



Analysis of the graph clearly shows an upward trend for the maximum total sales in the period between 2013 and 2015. However, a slight decline occurred in 2016.

Monthly Sales Analysis and Monthly Ranking with Offset and Rank

OFFSET is a data analysis function (DAX) in Power BI that, although similar in some ways to window functions, differs in its main purpose and use. Like window functions, OFFSET provides access to specific data within a table or matrix, but its approach is more oriented toward selecting a single value from a relative position.

In the example provided, it is evident how OFFSET can be used in a DAX calculation to obtain specific results. In the 'OFFSET Month' code presented, the OFFSET function is used to retrieve the value of the previous month's total sales. This is made possible by the coordinates specified in the OFFSET argument, where a -1 month shift from the current context is specified. In addition, it is important to note the use of other DAX functions, such as CALCULATE, ALLSELECTED, and ORDERBY, to manipulate the context of the data to achieve the desired result.

```
OFFSET Month = VAR SalesOffset = CALCULATE( [Total Sales], OFFSET(-1,ALLSELECTED('Dimension Date'[Calendar Month Number],
'Dimension Date'[Month]), ORDERBY('Dimension Date'[Calendar Month Number],ASC))) VAR Result= IF(NOT ISBLANK(SalesOffset),
[Total Sales] - SalesOffset) RETURN Result
```

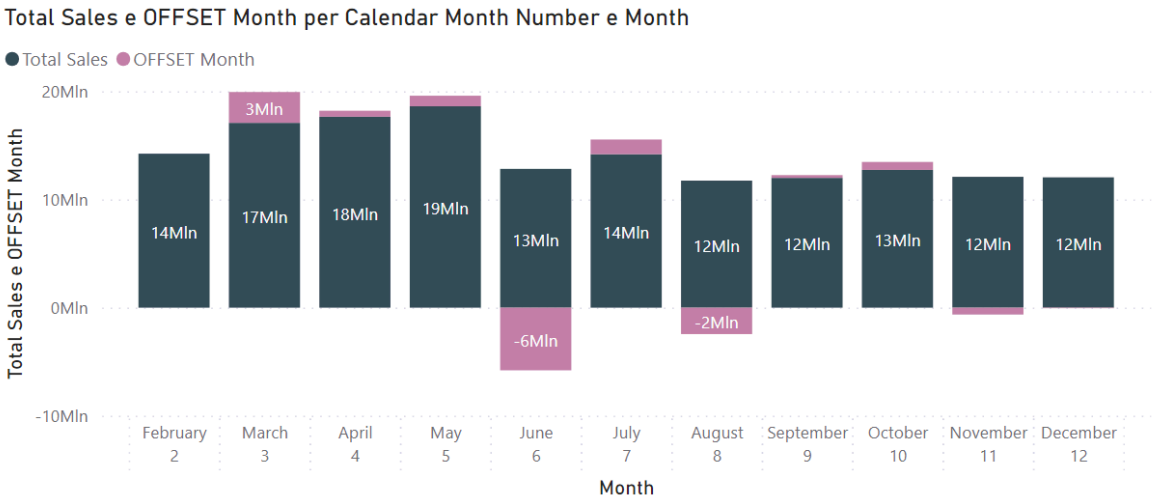
Another window function used for this analysis is the Rank function, whose main function is to calculate the position of a value within a data set sorted by a given expression. This is often used to determine the rank or ranking of an element in a data set. The general syntax of RankX is as follows: RANKX(table, expression, [value], [order], [ties]).

- table: The table on which to calculate the rank.
- expression: The expression to be evaluated to calculate the rank.
- value (optional): The value for which the rank is to be calculated. If omitted, the current element in context will be used.
- order (optional): Specifies the order in which to calculate the rank, which can be ASC (ascending) or DESC (descending). By default, the order is ASC.
- ties (optional): Specifies how to handle the positions of equal values.

During the analysis, I exploited the concept of Rank to calculate the rank of the data based on the "OFFSET Month" measure with respect to months. In other words, I am determining the rank of months based on monthly sales considering the previous month.

```
Rank =
VAR SalesOffset = CALCULATE([Total Sales], OFFSET(-1, ALLSELECTED('Dimension Date'[Calendar Month Number],'Dimension Date'
[Month]), ORDERBY('Dimension Date'[Calendar Month Number], ASC)))
RETURN
IF(
    NOT ISBLANK(SalesOffset),
    RANKX(ALLSELECTED('Dimension Date'[Calendar Month Number], 'Dimension Date'[Month]), [OFFSET Month]),
    BLANK()
)
```

Calendar Month Number	Month	Total Sales	OFFSET Month	Rank
1	January	16.817.501,05		
2	February	14.234.455,55	-2.583.045,50	11
3	March	17.084.624,55	2.850.169,00	1
4	April	17.643.985,45	559.360,90	5
5	May	18.617.646,30	973.660,85	3
6	June	12.828.369,00	-5.789.277,30	12
7	July	14.185.825,10	1.357.456,10	2
8	August	11.743.687,15	-2.442.137,95	10
9	September	11.997.243,15	253.556,00	6
10	October	12.732.968,30	735.725,15	4
11	November	12.095.864,70	-637.103,60	9
12	December	12.046.586,95	-49.277,75	8
Totale		**172.028.757,25**	**12.046.586,95**	**1**



Here are the results obtained: it should be noted that for the month of "January" it is not possible to calculate the OFFSET value, since it is the first month of the year. For a clearer visualization, I have included a graph that highlights how the first months are the most profitable. In particular, March stands out as having the highest OFFSET, with an increase of 3 million over February. On the other hand, June represents the month with the highest losses, registering a total difference of 6 million compared to May."

Conclusion

The analysis of this Power BI report, gave us a view of our business operations. We explored sales patterns, peak production periods, profitable regions, and frequently purchased items. These results are valuable for strategic decision making. Finally, the use of window functions and OFFSETs proved critical to conducting advanced analysis.